Ehren Schindelar
RIN 661677600
Sep. 23, 2018

Q1

This program question1.py attempts to find a hash collision between two provided text files, GoodText.txt and BadText.txt. One provides correct information, while the other provides false information. These must be formatted correctly for it to cover 2^{16} possibilities, without changeing the meaning of the text. It tries to find a collision over a 32-bit version of the MD5 hashing algorithm.

Q2

10.12
Right side values for $x = 1...10$:
$8, 5, 3, 9, 4, 8, 4, 9, 7, 4 = (3, 4, 5, 7, 8, 9)$
Values of y^2 for $y = 1...10$:
$1, 4, 9, 5, 3, 3, 5, 9, 4, 1$
List of valid points:
$(2, 4), (2, 7), (3, 5), (3, 6), (4, 3), (4, 8), (8, 3), (8, 8), (5, 2), (5, 9), (7, 2), (7, 9), (10, 2), (10, 9)$

10.13
$17/2 = 8.5$, mirror points over $y = 8.5$:
P: $(5, 8) \rightarrow (5, 9)$
Q: $(3, 0) \rightarrow (3, 17) \equiv (3, 0)$
R: $(0, 6) \rightarrow (0, 11)$

10.14
$E(1, 6) = y^2 = x^3 + x + 6(mod11)$
$G = (2, 7)$

$$\lambda = (3 * (2)^2 + 1)/(2 * 7) = 13/14 = 2/3 = 2 * (1/3)(mod11)$$
$$1/3^{-1}mod11 = 4 \text{ since } 4 * 3 = 12 = 1(mod11)$$
$$\lambda = 8(mod11)$$
$$x = \lambda^2 - x_1 - x_2modp$$
$$y = \lambda(x_1 - x) - y1modp$$
$$2G = (5, 2)$$
$$\lambda = (y_2 - y_1)/(x_2 - x_1) = -5/3 = 6/3 = 4 * 6 = 2(mod11)$$
$$3G = (8, 3)$$
$$\lambda = 7/6 = 7 * 2 = 3(mod11)$$
$$4G = (10, 2)$$
$$\lambda = 6/8 = 6 * 7 = 9(mod11)$$
$$5G = (3, 6)$$
$$\lambda = 10/1 = 10$$
$$6G = (7, 9)$$
$$\lambda = 2/5 = 2 * 9 = 7(mod11)$$

$$7G = (7, 2)$$
$$\lambda = 6/5 = 6 * 9 = 10(mod11)$$
$$8G = (3, 5)$$
$$\lambda = 9/1 = 9$$
$$9G = (10, 9)$$
$$\lambda = 2/8 = 2 * 7 = 3(mod11)$$
$$10G = (8, 8)$$
$$\lambda = 1/6 = 2(mod11)$$
$$11G = (5, 9)$$
$$\lambda = 2/3 = 2 * 4 = 8(mod11)$$
$$12G = (2, 4)$$
$$\lambda = 8/0$$
$$13G = (2, \infty)$$

10.15
a)
The public key would be the point $7G = (7, 2)$
b)
$C_m = [kG, P_m + kP_b] =$ [(8, 3), (10, 9) + (7, 2)]
$\lambda = 4/8 = 6(mod11)$
$C_m = [(8, 3), (8, 3)]$
c)
B can compute P_m by using the private key.
$P_m + kP_b - n_B(kG) = P_m + k(n_BG) - n_B(kG)$
Since $P_b = n_BG$.
Also, $k(n_BG) = n_B(kG)$ so they can compute P_m by finding:
$P_m = C_m[1] - n_b * C_m[0] = (8, 3) - 3 * (8, 3)$

Q3

I made two python programs that emulate the Miller-Rabin and Pollard-Rho algorithms. The only one of the provided numbers that was found to be not prime was 520482, which is obviously even. The others are probably prime. Using the Pollard-Rho algorithm, I found that the prime factorization of 520482 is $2 * 3 * 223 * 389$.